AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Aerospace Sciences Meeting

10.2514/6.2018-1101

Customized Grid Generation Codes for Benchmark Three-Dimensional Flows

Hiroaki Nishikawa[*] and Boris Diskin[†]
National Institute of Aerospace, Hampton, VA 23666

Customized grid generation and coarsening codes have been developed for three-dimensional benchmark configurations as part of NASA's Turbulence Modeling Resource website, namely, a hemisphere cylinder, and the ONERA M6 wing. The codes have been used to generate families of consistently refined nested grids with different element types. Fine grids of high density with up to 400 million degrees of freedom have been generated. These grids are for use in computing reference solutions for verification of advanced solver technologies for complex turbulent flows. This paper presents details of the grid generation and coarsening processes, describes capabilities and customized features built in the codes, and demonstrates superior grid quality metrics such as skewness and face planarity.

I. Introduction

Grid generation is an important component of computational fluid dynamics (CFD) analysis. The accuracy of discrete solutions and their ability to adequately predict complex turbulent flows are significantly affected by grid characteristics such as resolution, element type, anisotropy, regularity, skewness, face planarity, etc. Convergence of many iterative methods also depends on the grid characteristics. Generation of suitable grids for complex flows is often a trial and error process that may take significantly longer time than flow simulations conducted on those grids.

In many studies, a family of grids is needed to assess accuracy and increase confidence in flow simulations. Accuracy of computational solutions is often established through grid convergence studies. Current guidelines [1, 2] for grid convergence studies emphasize a parametric similarity of grids forming a family and an asymptotic convergence order, which is expected to be observed on three-to-four fine grids in a family. The grid resolution required for establishing a convergence order is sought through a consistent grid refinement. For structured grids, a family of consistently refined grids is typically derived recursively, starting from the finest grid. Each coarser grid in the family is derived from the preceding finer grid by removing every-other grid plane/line in each dimension. For general unstructured grids, generation of families of consistently refined grids remains a challenge.

Collaborative studies that involve aerodynamic flows critically rely on the ability to efficiently generate consistent families of grids, including high-resolution grids, for distribution among a diverse pool of participants. Multiple international collaborative efforts have been initiated to assess and advance predictive capabilities of computational tools used for aerospace applications. Many workshops sponsored by the American Institute of Aeronautics and Astronautics (AIAA), such as the Drag Prediction Workshops [3] and the High-Lift Prediction Workshops [4], adopted the collaboration model, in which grids are generated and stored in a designated location to be downloaded and used by contributors from different organizations across the globe. While this collaboration model is acceptable for studies that involve relatively small grids, it is not practical for high-resolution grids with billions of degrees of freedom. For such grids, the storage requirements and the download time are not reasonable. A more effective approach is to distribute a grid generation package that can generate the same grids in different computational environments.

Various commercial and non-commercial grid generation software packages have been developed and are widely used [5, 6, 7, 8]. Many of these packages are designed to address challenges of practical applications

[*] Associate Research Fellow (hiro@nianet.org), Associate Fellow AIAA
[†] Research Fellow (bdiskin@nianet.org), Associate Fellow AIAA

and have complex user interfaces to facilitate handling of various geometries and grid types and tailoring for different flow conditions. In collaborations that involve dozens of participants, uniformity of inputs used for generation of presumably the same grids in different environments is difficult to achieve. Also, general grid generation packages may not be efficient for benign geometries or universally available. A package designated for a specific geometry may significantly reduce the grid generation time and improve grid characteristics. Detailed access to geometry data, close control of the grid generation process, and familiarity with the needs of the study and involved solvers allow many additional benefits. The benefits include generation of families of grids with different elements that have the same point distribution, the same shape and position for the far-field boundary used across all families, generation of line information, multigrid data structures including intergrid interpolation and restriction operators, in-situ grid quality control (positive control volumes, element aspect ratio, skewness, face planarity, etc.), optimal grid partition for solvers that may use lines among others.

The Turbulence Modeling Resource (TMR) website [9,10] established at NASA Langley Research Center provides the aerodynamics community with benchmark cases for verification and validation of turbulence models. Recently, a section Cases and Grids for Turbulence Model Numerical Analysis has been added to the TMR website. The test cases in this section are designed primarily for numerical analysis of solver technology used in turbulent flow simulations, e.g., grid and iterative convergence properties, effects of specific discretizations, grid-refinement strategies, etc. In support of the TMR objectives, special sessions on solver technology for turbulent flows were held at the AIAA Science and Technology Forum and Exposition (SciTech) conferences in 2015 and 2016. Advanced solver technologies for the Reynolds-Averaged Navier Stokes (RANS) equations were demonstrated in application to relatively simple benchmark flows in two dimensions (2D) and three dimensions (3D). For each benchmark flow, the reference solutions have been computed on families of consistently refined grids with high-resolution, fine grids intended to establish grid converged solutions. Detailed descriptions of the reference solutions have been posted on the TMR website. For those simple benchmark flows, it was feasible to store all grids in one location (at the TMR website) and request all contributors to download the grids. Even then, a hybrid approach was used for some 3D configurations. In particular, the 3D bump grid generation process required to download a 2D bump grid and apply a program that expands the grid for 3D simulations.

In the quest to provide benchmark cases that are representative of realistic turbulent aerodynamic flows, two 3D configurations have been suggested for a special session at SciTech 2018. A 3D Hemisphere Cylinder (HC) configuration provides test cases for turbulent flows over a smooth body of revolution. The geometry is taken from a report [11] describing the corresponding experimental model. The 3D ONERA M6 (OM6) wing configuration provides test cases for transonic turbulent flows over a swept wing. The geometry definition follows recent ONERA papers [12, 13, 14]. ONERA created a CAD geometry for the wing with a sharp trailing edge for the purposes of this CFD exercise, as described in Refs. [12,14]. Customized grid generation and coarsening codes have been developed for the HC and OM6 configurations and used by contributors to the SciTech 2018 special session. The codes generate families of high-quality consistently refined grids with different elements. These grids are used to establish reference solutions through a grid convergence study reported in an accompanying paper [15].

This paper presents details of grid generation and coarsening procedures implemented in these codes and describes the customized built-in features of the codes. The material of the paper is organized in the following order. In Section II, the code distribution and input parameters are described. In Section III, the grid generation process for the HC configuration, which is analytically defined, is described, including methods to avoid the polar singularity typical for structured grids. Specific features such as generation of nested coarse grids, grid output format options, line generation, line-preserving domain decomposition, and inter-grid interpolation are also presented. In Section IV, the grid generation for the OM6 configuration is achieved through the mapping between the HC and OM6 configurations. In Section V, grid characteristics are discussed for HC and OM6 grids. In Section VI, the parameters of the input files are explained and representative grids are shown. In Section VII, concluding remarks are given.

II. Distribution

The latest versions of the HC and OM6 grid generation packages are available at the TMR website [9]:

`hc_v8p4_release.tar.gz`	`Hemisphere-cylinder grid generation package`
`wing_v4p4_release.tar.gz`	`OM6 wing grid generation package`

Each package contains a grid generation code, a grid coarsening code, which is common for both packages, sample input files, and an instruction file (`readme_release.txt`). The hemisphere-cylinder package contains:

`hcf_hc_v8p1.f90`	Grid generation code
`hcf_coarsening_v3p4.f90`	Grid coarsening code
`input.nml_strct`	Sample input file for generating a structured grid
`input.nml_tetra`	Sample input file for generating a tetrahedral grid
`input_coarsen.nml_strct`	Sample input file for coarsening the structured grid
`input_coarsen.nml_tetra`	Sample input file for coarsening the tetrahedral grid
`readme_release.txt`	Instruction

The OM6 wing package contains:

`hcf_wing_v4p1.f90`	Grid generation code
`hcf_coarsening_v3p4.f90`	Grid coarsening code
`om6_wing_section_sharp.dat`	OM6 wing section data [14]
`input.nml_strct`	Sample input file for generating a structured grid
`input.nml_tetra`	Sample input file for generating a tetrahedral grid
`input_coarsen.nml_strct`	Sample input file for coarsening the structured grid
`input_coarsen.nml_tetra`	Sample input file for coarsening the tetrahedral grid
`readme_release.txt`	Instruction

The grid generation codes use the FORTRAN namelist feature, and read the input file named `input.nml`; a sample grid generation input file needs to be renamed when used. The grid coarsening code reads the input file named `input_coarsen.nml`; the sample coarsening input files also need to be renamed. The sample input files provided with the packages generate small-size sample grids that do not require significant memory. The input files to generate large-size grids recommended for the SciTech 2018 special session are separately available at the TMR website. Girds with different element types can be generated by changing a single parameter in the input file. Input parameters for the HC and OM6 grid generation codes, and the grid coarsening code are listed and explained below. For further details, see the source codes, where all available namelist variables are shown and explained.

Input parameters for the HC grid generator:

`target_reynolds_number`	=	Freestream Reynolds number based on the diameter of the hemisphere.
`target_y_plus`	=	Desired y^+ value.
`igrid_type`	=	1: prism, 2: tetra, 3: mixed (prism/tetra), 4: mixed(prism/hex), 5: structured
`R_outer`	=	Distance to the outer boundary.
`nnodes_cylinder_input`	=	Number of elements along the cylinder.
`nr_gs`	=	Number of elements from the tip to the hemisphere base.
`nre`	=	Number of elements along the grid line from the surface to the outer boundary.
`domain_cut`	=	1: full geometry, 2: half geometry
`generate_ugrid_file`	=	T: write .ugrid file, F: do not write.

`ugrid_file_unformatted` = T: binary .ugrid file, F: ASCII .ugrid file.
`generate_k_file` = T: write .k file, F: do not write.
`generate_line_file_nc` = T: write node line files, F: do not write.
`generate_line_file_cc` = T: write element line file, F: do not write.
`generate_tec_file_b` = T: write Tecplot boundary file, F: do not write.
`generate_tec_file_v` = T: write Tecplot volume file, F: do not write.

Input parameters for the OM6 grid generator:

`target_reynolds_number` = Freestream Reynolds number based on the diameter of the hemisphere.
`target_y_plus` = Desired y^+ value.
`igrid_type` = 1: prism, 2: tetra, 3: mixed (prism/tetra), 4: mixed(prism/hex), 5: structured
`airfoil_type` = 0: Use discrete airfoil data, 1: Use NACA00XX series.
`airfoil_data_file` = Filename of a discrete airfoil data.
`tr` = Taper ratio
`beta` = Sweep angle
`R_outer` = Distance to the outer boundary.
`nnodes_cylinder_input` = Number of elements along the cylinder.
`b` = Semi span
`nr_gs` = Number of elements from the tip to the hemisphere base.
`nre` = Number of elements along the grid line from the surface to the outer boundary.
`wing_side` = 1: left wing, 2: right wing
`root_le_x` = x coordinate of the LE at the root.
`stretching_tanh_towards_lete` = Stretching factor to cluster nodes at LE and TE.
`generate_p3d_ufmt_file` = T: Write .p3d/.umft and .nmf files (`igrid_type` = 5 only), F: do not write.
`generate_ugrid_file` = T: write .ugrid file, F: do not write.
`ugrid_file_unformatted` = T: binary .ugrid file, F: ASCII .ugrid file.
`generate_k_file` = T: write .k file, F: do not write.
`generate_line_file_nc` = T: write node line files, F: do not write.
`generate_line_file_cc` = T: write element line file, F: do not write.
`generate_tec_file_b` = T: write Tecplot boundary file, F: do not write.
`generate_tec_file_v` = T: write Tecplot volume file, F: do not write.

Input parameters for the grid coarsening code:

`project` = `hc_prism`, or `hc_tetra`, `hc_mixed`, `hc_mixed_ph`, `hc_strct`,
`wing_prism`, `wing_tetra`, `wing_mixed`, `wing_mixed_ph`, `wing_strct`,
`tet2prism` = T: prismatic coarse grids, F: tetrahedral coarse grids. (for `hc_tetra` and `wing_tetra`).
`ugrid_file_unformatted` = T: binary .ugrid file, F: ASCII .ugrid file.
`k_file_unformatted` = T: binary .k file, F: ASCII .k file.
`rmp_file_unformatted` = T: binary .rmp file, F: ASCII .rmp file.
`generate_tec_file_b` = T: write Tecplot boundary file, F: do not write.
`generate_tec_file_v` = T: write Tecplot volume file, F: do not write.

III. Grid Generation for Hemisphere Cylinder Configuration

III–A. Geometry and computational domain

The full hemisphere cylinder geometry consists of a hemisphere and a cylinder with the radius of 0.5 inch. The total length of the configuration is determined by the input parameter `x2`. See Figure 1(a). The radius has been chosen because the model used in the experiment reported in Ref.[11] had the radius of 0.5 inch and the length of 10 inches. The apex of the hemisphere is located at the origin. The computational domain is bounded by the hemisphere-cylinder surface, a farfield hemisphere with a user-specified radius, `R_outer`, and an outflow plane located at $x =$`x2`. Optionally, a half geometry can be used by assigning symmetry boundary conditions at the $y = 0$ plane as shown in Figure 1(b).

III–B. Grid types

Table 1: Summary of grid types for hemisphere cylinder configuration. The symmetry boundary exists only for half geometry.

Grid Type	Hemisphere Surface	Cylinder Surface	Farfield Surface	Outflow Plane	Symmetry Plane	Polar Singularity	Hemisphere Volume	Cylinder Volume
1 (prism)	Tria	Tria	Tria	Quad	Quad	N/A	Prism	Prism
2 (tet)	Tria	Tria	Tria	Tria	Tria	N/A	Tet	Tet
3 (prism/tet)	Tria	Tria	Tria	Quad/Tria	Quad/Tria	N/A	Prism/Tet	Prism/Tet
4 (prism/hex)	Tria	Quad	Quad/Tria	Quad	Quad	N/A	Prism	Hex
5 (struct)	Quad/Tria	Quad	Quad/Tria	Quad	Quad	At apex	Prism/Hex	Hex

Four types of unstructured grids and one type of structured grids can be generated by the current codes as summarized in Table 1. Structured grids are mostly composed of hexahedral elements. Any interior node of a structured grid is generally connected to the same number of grid elements. The nodes and the elements can be addressed with the indices (i, j, k). Structured grids generated for the hemisphere-cylinder configuration have a polar singularity at the apex of the hemisphere, i.e., the hexahedral elements attached to the polar axis $(x \leq 0, y = 0, z = 0)$ degenerate to prismatic elements. The nodes on the polar axis connect to many grid elements; the number of elements connected to a typical polar-axis node increases on finer grids. For consistency of structured-grid index references, multiple nodes with different indices and identical coordinates are preserved at the polar axis.

Unstructured grids generated for the hemisphere-cylinder configuration do not have polar singularity. The grids can be composed from elements of different types, e.g., tetrahedra, hexahedra, prisms. Nodes may connect to different number of grid elements, and the elements can be of different types. Unstructured grids generally cannot be referenced by indices, such as (i, j, k). However, unstructured grids generated by the current codes are regular; each node can be referenced by an extended set of indices $(k_1, k_2, k_3, k_4, k_5)$ defined in Sections III–C, III–D, and III–G.

III–C. Hemisphere surface mesh

The hemisphere surface mesh generation is controlled by the input parameter `nr_gs`, which indicates the number of edges running along a grid line from the apex to the base perimeter of the hemisphere. The same surface mesh is generated for all unstructured grid types. A regular triangular-sector mesh is generated first as in Figure 2(a); the sector has $\texttt{nr\_gs}^2$ triangular faces. Then, six copies of this sector are generated and put together to form a disk as in Figure 2(b). The disk is mapped onto the hemisphere as in Figure 2(c). Finally, the mesh is rotated by 90° around the x-axis as in Figure 2(d), so that the $y = 0$ plane can be used to generate a half-geometry grid if requested. As a result, the full-geometry hemisphere mesh has $6\times$ `nr_gs` edges along the base circle and the total of $6\times$ $\texttt{nr\_gs}^2$ triangular surface faces. The node at the apex has exactly six neighbor nodes; this is independent of the mesh size. By the construction, the nodes of the resulting surface mesh can be addressed by a set of indices (k_1, k_2, k_3), $k_1 \in [-\texttt{nr\_gs}, \texttt{nr\_gs}]$, $k_2 \in [-\texttt{nr\_gs}, \texttt{nr\_gs}]$, and $k_3 = -k_1 - k_2$. See Figure 3.

For structured grids, nodes are first placed along the base circle in the same positions as for the corresponding unstructured grids. Mesh lines with nr_gs intervals are generated from the apex to base-circle nodes. The resulting structured surface mesh shares with the unstructured counterparts the node at the apex, all nodes along the base circle, and nodes along the six lines corresponding to the triangular-sector boundaries. Also, in this way, the surface meshes of all grid types may share exactly the same node distribution over the cylinder surface. See Figure 4 for indices used to represent the structured grid.

Structured full-geometry hemisphere meshes are axisymmetric meshes with quadrilateral faces almost everywhere except near the apex of the hemisphere, where quadrilateral faces degenerate into triangular ones. The total number of surface faces is the same as for the corresponding unstructured hemisphere surface mesh, $6\times \texttt{nr\_gs}^2$. The unstructured surface mesh is not axisymmetric, but is periodic with respect to a 60° circumferential rotation and reflection symmetric with respect to planes with circumferential angles that are multiples of 30°. The circumferential angle is computed as $\phi = \cos^{-1}\left(z/\sqrt{y^2+z^2}\right)$.

III–D. Cylinder surface mesh

The surface mesh over the cylinder is generated from the hemisphere base circle. Nodes from the hemisphere base circle are mapped on the parallel cylinder circles located along the cylinder surface all the way to the cylinder base circle. The number of cylinder circles is regulated by the input parameter, nnodes_cylinder_input, which does not account for the hemisphere base circle. In other words, nnodes_cylinder_input is the number of edges in a grid line along the cylinder from a hemisphere base circle node to the corresponding node at the end of the cylinder. An exponential stretching is applied to the x-coordinate of cylinder circles for smooth transition across the hemisphere-cylinder junction. The first x-direction cylinder-mesh spacing is required to be 10% larger than a representative spacing in the hemisphere mesh, i.e., $(\pi/4)/\texttt{nr\_gr}$. The stretching parameter is computed numerically to meet this requirement. A cylinder circle on the cylinder surface is represented by the index k_4. For unstructured grids, $k_4 \in [0, \texttt{nnodes\_cylinder\_input}]$, where $k_4 = 0$ over the hemisphere surface. For structured grids, $k_4 \in [0, \texttt{nr\_gs} + \texttt{nnodes\_cylinder\_input}]$ and $k_4 = 0$ at the apex node. See Figures 3 and 4.

Surface faces of structured grids are quadrilateral; surface faces of unstructured grids are either quadrilateral or triangular. Triangular faces are generated by a diagonal division of quadrilateral surface faces. In the current code, the surface normal vectors, (n_x, n_y, n_z), are computed analytically at each node. The normal vectors at the hemisphere base circle are slightly deflected towards the negative x-direction to improve grid smoothness. The normal vectors are used for volume-grid generation including a farfield-boundary surface mesh.

III–E. Farfield boundary

The farfield boundary is a hemisphere of radius, R_outer. The hemisphere-cylinder surface mesh is mapped onto the farfield-boundary hemisphere. Each node on the hemisphere-cylinder surface is mapped by using the index i that runs from the apex of the hemisphere ($i = 1$) to the cylinder base ($i = \texttt{nr\_gs} + \texttt{nnodes\_cylinder\_input} + 1$). Surface nodes with the same index i are located on a circle in the (y, z)-plane centered at the x-axis. The coordinates of the corresponding node on the farfield boundary are determined as

$$x = \texttt{x2} - \texttt{R\_outer}\cos(\theta), \quad y = \texttt{R\_outer}\sin(\theta)\, n_y, \quad z = \texttt{R\_outer}\sin(\theta)\, n_z, \tag{1}$$

where n_y and n_z are the y- and z-components of the unit normal vector at a hemisphere-cylinder surface node with index i, and

$$\theta = (i-1)\Delta\theta, \quad \Delta\theta = \frac{0.5\pi}{\texttt{nr\_gs} + \texttt{nnodes\_cylinder\_input}}. \tag{2}$$

Faces at the farfield boundary are defined by mapping each node into the connectivity list available in the hemisphere-cylinder surface mesh.

III–F. Boundary layer nodes

The codes generate grids for viscous-flow simulations. An anisotropic boundary-layer grid is generated given a target Reynolds number and a $y+$ value specified through the input parameters, target_reynolds_number and `target_y_plus`, respectively. Following Ref. [16], we compute the first off-the-wall distance, Δr_1, as

$$\Delta r_1 = \frac{\texttt{target\_y\_plus}\sqrt{2/c_f}}{\texttt{target\_reynolds\_number}}, \quad c_f = 0.026 \; \texttt{target\_reynolds\_number}^{-\frac{1}{7}}. \tag{3}$$

The boundary layer thickness, $r_{\max}$, is estimated by a formula for a turbulent boundary layer over a flat plate [17]:

$$r_{\max} = (\texttt{bl\_height\_factor})(0.37)(\texttt{x2}) \; \texttt{target\_reynolds\_number}^{-\frac{1}{5}}, \tag{4}$$

where `bl_height_factor` is another input parameter with the default value $\texttt{bl\_height\_factor} = 0.5$. Then, the largest spacing across the boundary layer grid is defined as

$$\Delta r_m = \max\left(2\Delta r_1, \frac{0.025\pi}{\texttt{nr\_gs}}\right). \tag{5}$$

Given Δr_1 and Δr_m, a geometric series,

$$\Delta r_k = g^{k-1}\Delta r_1, k = 1, 2, \cdots, m, \tag{6}$$

is constructed, such that $g^{m-1}\Delta r_1 = \Delta r_m$:

$$m = \left\lceil \frac{\log(g'\Delta r_m/\Delta r_1)}{\log(g')} \right\rceil, \quad g' = \frac{r_{\max} - \Delta r_1}{r_{\max} - \Delta r_m}, \tag{7}$$

$$g = \left(\frac{\Delta r_m}{\Delta r_1}\right)^{\frac{1}{m-1}}, \tag{8}$$

where $\lceil \cdot \rceil$ denotes the ceiling function, m is the number of edges (thus, the number of nodes is $m+1$) within the boundary layer and g is the geometric factor.

The volume grid inside the boundary-layer region is designed to be orthogonal to the hemisphere-cylinder surface. Input parameter, `nre` defines the total number of edges along a grid line from the hemisphere-cylinder surface to the farfield boundary. A grid cannot be generated if $\texttt{nre} \le m$; in such a case, the code stops with a warning that `nre` should be increased. If $\texttt{nre} > m$, then m off-surface advancing grid layers are generated within the boundary layer. Nodes are placed along each grid line attached to the hemisphere-cylinder surface according to the geometric series defined by Eq. (6). The grid line is colinear with the normal vector defined at the surface node.

III–G. Volume grid nodes outside of boundary layer

Grid lines extend beyond the boundary layer toward the farfield boundary. Each grid line connects a node on the hemisphere-cylinder surface with its counterpart node on the farfield hemisphere constructed according to Eq. (1). Beyond the boundary layer, advancing layers of nodes are generated with an exponential stretching toward the boundary-layer region. The stretching factor is determined iteratively to ensure a smooth transition from the boundary layer region to the outer region. The first outer-region grid spacing is required to be Δr_m, which is the last grid spacing within the boundary layer.

The line direction changes gradually from the surface normal direction to the direction toward the node on the farfield hemisphere. Assuming a thin boundary layer, the latter direction is closely approximated by the direction from the first grid-line node on the hemisphere-cylinder surface to the last grid-line node on the farfield hemisphere. See Figure 5, where the surface normal and end-to-end direction are denoted by vectors $\mathbf{n}_1$ and $\mathbf{n}_2$, respectively. The blending of the two directions is performed by the exponential function that is reciprocal to the one used for the node placement, so that the direction quickly switches from $\mathbf{n}_1$ to $\mathbf{n}_2$.

For both structured and unstructured grids, index k_5 is used to refer to advancing layers of nodes or, in other words, to address a node on a grid line connecting the hemisphere-cylinder surface with the the farfield boundary. See Figures 3 and 4. The entire index information $(k_1, k_2, k_3, k_4, k_5)$ is written in a file with the extension `.k`. This information is used by the grid coarsening code.

III–H. Volume elements

After the advancing layers of grid nodes have been formed, various volume elements can be constructed by specifying connectivity of nodes located at consecutive advancing layers. The connectivity within each advancing layer is mapped from the surface-mesh node connectivity. Prismatic and hexahedral grid elements are formed by adding edges between nodes that are mapped to the same surface node and belong to neighboring advancing layers. Hexahedral elements can be divided into prismatic elements to provide a fully prismatic grids. All or some of prismatic elements can be further divided into tetrahedral elements to provide fully tetrahedral or mixed-element grids.

For unstructured grids, four grid types are available: fully prismatic grids are denoted as *prism*, fully tetrahedral grids are denoted as *tet*, mixed-element grids composed of prismatic and tetrahedral elements are denoted as *prism/tet*, mixed-element grids composed of hexahedral and prismatic elements are denoted as *prism/hex*. For structured grids, prismatic elements are generated around the polar axis; hexahedral elements are generated everywhere else. This type of grid is referred to as *struct*. In prim/tet grids, prisms are used inside the boundary layer region, and tetrahedra are used in the rest of the domain. In prism/hex grids, prisms are used over the hemisphere, and hexahedra are used over the cylinder. A summary of volume element types is given in Table 1.

Downloaded by Hiroaki Nishikawa on January 8, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-1101

III–I. Symmetry boundary

The grid generation code offers two geometry options, full and half geometries, which can be chosen by the input parameter `domain_cut`: `domain_cut`=1 for a full geometry and `domain_cut`=2 for a half geometry. The half-geometry configuration has a symmetry plane at $y = 0$, and a grid is generated only in the region $y \geq 0$. The full-geometry configuration has no symmetry plane and the grid is generated within the entire farfield hemisphere. The node and element distributions of half and full geometries match in the region $y \geq 0$. The element type of symmetry boundary faces is automatically derived from volume elements. See Table 1.

III–J. Outflow boundary

Outflow boundary is a plane located at $x = \texttt{x2}$. Similar to the symmetry plane, the element type of the outflow boundary faces is automatically derived from volume elements. See Table 1.

III–K. Featured Capabilities of Grid Generation Codes

III–K.1. Generation of families of consistent nested grids

A separate grid coarsening code is provided to generate a family of consistent grids. The grid coarsening code treats the grid generated by the grid generation code as the finest grid in the family, and performs regular coarsening by removing every other node in each dimension. Coarsening stops when the next coarse grid cannot be generated. The maximum number of coarse grids that can be generated for a given fine grid is given by

$$\min(m_1, m_4, m_5), \tag{9}$$

where

$$m_1 = \lfloor \log_2(\texttt{nr\_gs}) \rfloor, \quad m_4 = \lfloor \log_2(\texttt{nnodes\_cylinder\_input}) \rfloor, \quad m_5 = \lfloor \log_2(\texttt{nre}) \rfloor. \tag{10}$$

Parameters `nr_gs`, `nnodes_cylinder_input`, and `nre` are used for generation of the finest grid, and thus should be carefully chosen to allow a desired number of coarser grids. In the process of grid coarsening, the code generates a `.rmp` file that contains a list of removed fine-grid nodes for each coarse element. This file is generated for each level except for the finest level. For a tetrahedral grid, there is an option to generate prismatic coarse grids. The option is enabled by setting `tet2prism = T` in the input file.

The grid coarsening code requires certain files generated by the grid generation code: `.ugrid` file, `.k` file, and the line information files. Therefore, to allow generation of a grid family, the following setting must be used when the finest grid is generated:

$$\begin{aligned}\texttt{generate\_ugrid\_file} &= \text{T}\\ \texttt{generate\_k\_file} &= \text{T}\\ \texttt{generate\_line\_file\_nc} &= \text{T}\\ \texttt{generate\_line\_file\_cc} &= \text{T}\end{aligned}$$

Also, the file types (i.e., formatted or unformatted) must match between the input files for the grid generation and coarsening codes.

III–K.2. Line construction

Line information files are generated for all grids and can be used for line-implicit relaxation and/or line agglomeration. There are two versions of line files, one for node-centered formulations and another one for cell-centered formulations. For node-centered formulations, files with the extension `.lines_fmt_all` contain lines of nodes; each line starts from a node on the hemisphere-cylinder surface and lists, in order, all nodes of the line up to the last node on the farfield boundary. Other files with the extension `.lines_fmt` contain lines only within the boundary layer region ($m+1$ nodes per line). Line generation is controlled by the input parameter `generate_line_file_nc`. Setting `generate_line_file_nc` = T enables generation of lines of nodes.

Similarly, for cell-centered formulations, there are two types files with extensions `.lines_fmt_all_cc` and `.lines_fmt_cc`. The files contain lines of cells. Each line is an ordered list of face-connected cells. Prismatic and hexahedral cells form natural lines following the grid advancing layers. Lines of tetrahedral cells include all tetrahedral cells that fall withing a corresponding line of prismatic cells. Three tetrahedral cells conform to one prismatic cell. The files with the extension `.lines_fmt_all_cc` have lines that have the first cell with a boundary face on the hemisphere-cylinder surface and the last cell with a face on the farfield boundary. A line in the file with the extension `.lines_fmt_cc` has the last cell that is connected to the nodes at advancing layers m and $(m+1)$.

III–K.3. Inter-grid interpolation

Prolongation information files are generated for node- and cell-centered formulations. For a node-centered formulation, files with the extension `.prolong_nc_seq` contain the number of nodes and the number of nested nodes followed by the list of pairs of nested nodes in the format: fine-grid node, coarse-grid-node. Files with the extension `.prolong_nc` contain the number of nodes followed by a set of data: a fine-grid node and the number of associated coarse-grid nodes followed by the list of the coarse-grid nodes and interpolation coefficients. Note that the `.prolong_nc` files contain the nested nodes as well. For cell-centered formulations, files with the extension `.prolong_cc` contain the list of fine-grid elements agglomerated into each coarse-grid element. A suggested interpolation for the cell-centered formulation is injection. These prolongation files can be used for multigrid, grid sequencing, grid agglomeration, and load balanced domain decomposition.

III–K.4. Load-balanced line-preserving domain decomposition

A load-balanced line-preserving cell-centered domain decomposition can be achieved based on the surface mesh faces. The domain decomposition is naturally derived from the cell-centered line construction described in Section III–K.2 and the cell-centered prolongation operator described in Section III–K.3. Each line of cells conforms to a face on the hemisphere-cylinder surface. Four faces of a fine surface mesh conform to a face of a coarser surface mesh. The fine-grid partition includes all cells that fall within a coarse-grid line. Since all lines of the same grid have the same number of cells, all fine-grid partitions contain exactly the same number of cells as well. The minimum number of possible partitions is equal to the number of surface faces on the coarsest grid. For a grid level k, $(k_{max} - k)$ partition files are generated with the extension `ntrl_part_xx`, where `xx` indicates the number of partitions, where k_{max} is the coarsest level, and $k = 1, 2, \ldots, k_{max} - 1$. By construction, the number of partitions is equal to the number of surface-grid cells in a coarse grid. The partition file has the total number of cells followed by a single column of data, indicating the partition number for each cell of the grid.

IV. Grid Generation for the OM6 Wing Configuration

A code that generates grids over a wing is available at the TMR website. The wing grids are topologically equivalent to the hemisphere-cylinder grids. The wing grid generation code generates a full-geometry hemisphere-cylinder surface grid by the same algorithm used in the hemisphere-cylinder grid generation code, and maps it onto a wing geometry. The code can generate a grid for the NACA00XX family of wing sections and for a general wing section defined by discrete data. This wing grid generation code is used to generate grid families for the OM6 wing configuration.

IV–A. Geometry and computational domain

The OM6 wing geometry with a sharp trailing edge is defined in Ref. [14]; a sketch is shown in Figure 6. The wing section data set given in Ref. [14] and a cubic spline are used to generate the section profile. The wing section data is stored in the file `om6_wing_section_sharp.dat`. The code reads the data from a file specified by the input parameter `airfoil_data_file = ‘‘om6_wing_section_sharp.dat"`. The wing can be left or right; the input parameter `wing_side` = 1 corresponds to the left wing, and `wing_side` = 2 corresponds to the right wing. Figure 6 shows the right wing configuration. The location of the wing is controlled by the input parameter `root_le_x`, which defines the x-coordinate of the leading edge location at the root. The taper ratio, sweep angle, and semi-span are specified by the input parameters `tr`, `beta`, and `b`, respectively. The wing root chord is set to be one in the grid generation code. Therefore, `tr` is equivalent to the tip chord.

For the OM6 wing with a unit root chord, the following values are set:

$$\begin{aligned} \texttt{tr} &= 0.5625159852668158, \\ \texttt{beta} &= 29.9990, \\ \texttt{b} &= 1.476017976219800. \end{aligned} \tag{11}$$

The outer boundary is taken as a hemisphere centered at the origin.

IV–B. Mapping

In the mesh generation process, a surface mesh for a full-geometry hemisphere-cylinder configuration is generated first. At this point, the hemisphere cylinder configuration is defined as in Figure 1. This surface mesh is then mapped onto a wing: the cylinder is mapped onto the main wing and the hemisphere is mapped onto the wing tip. Each surface node is projected onto the airfoil section profile. Over the cylinder part, the projection is performed along the z-direction in the plane perpendicular to the x-axis. The airfoil section is represented by a cubic spline constructed from the discrete section data with the chordwise distance s from the leading edge ($y = 0.5$, $z = 0$) towards the trailing edge ($y = -0.5$, $z = 0$) as a parameter. The node with coordinates $(x, y, z) = (x_p, y_p, z_p)$ is projected onto the airfoil with a new z-coordinate determined by the cubic spline for $s = 0.5 - y_p$. Over the hemisphere, the projection is performed in the plane defined by the node to be projected and the line through the center of the hemisphere-base circle that is colinear to the y-axis. The projection defines the half-thickness of the section, which is sufficient for a symmetric section. Then, the wing is tapered with the input parameter `tr`, and the wing sections are relocated along the y-axis to meet the requested sweep angle `beta`. Once the surface mesh is mapped, the interior nodes and elements are generated as described earlier for the hemisphere-cylinder grid. Finally, the grid is rotated around the z-axis and translated along the y-axis such that the symmetry plane coincides with the (x, z)-plane at $y = 0$, and all nodes have non-negative or non-positive y-coordinates for the right and left wing configurations, respectively. Note that the outflow boundary for the hemisphere-cylinder configuration is used as a symmetry boundary plane in the wing configuration.

IV–C. Additional Features

As OM6-wing grids are topologically similar to the corresponding hemisphere-cylinder grids, all features described for the hemisphere cylinder configuration are available to the OM6 wing grids. In particular, given a fine grid, the grid coarsening code generates consistent families of nested grids for the OM6 wing

Table 2: Summary of grid types for the OM6 wing configuration.

Grid Type	Hemisphere Surface	Cylinder Surface	Farfield Surface	Symmetry Plane	Polar Singularity	Hemisphere Volume	Cylinder Volume
1 (prism)	Tria	Tria	Tria	Quad	N/A	Prism	Prism
2 (tet)	Tria	Tria	Tria	Tria	N/A	Tet	Tet
3 (prism/tet)	Tria	Tria	Tria	Quad/Tria	N/A	Prism/Tet	Prism/Tet
4 (prism/hex)	Tria	Quad	Quad/Tria	Quad	N/A	Prism	Hex
5 (struct)	Quad/Tria	Quad	Quad/Tria	Quad	At tip	Prism/Hex	Hex

configuration. A feature unique to OM6-wing grids is the ability to control the grid stretching towards the leading and trailing edges. The stretching is performed by a hyperbolic tangent function with the input parameter `stretching_tanh_towards_lete` used as a stretching factor. The default value is set to be `stretching_tanh_towards_lete` = 4.5. A value of `stretching_tanh_towards_lete` close to zero corresponds to no stretching. A larger value clusters nodes towards the leading and trailing edges.

V. Grid characteristics

V–A. Hemisphere Cylinder Grids Characteristics

Table 3: Statistics of four finest grids for hemisphere-cylinder grid families.

Grid	Family 2 (tet)		Family 4 (prism/hex)			Family 5 (struct)		
	Tetrahedra	Nodes	Hexahedra	Prisms	Nodes	Hexahedra	Prisms	Nodes
L1	424,673,280	71,368,353	62,914,560	15,728,640	71,368,353	78,397,440	245,760	79,115,841
L2	53,084,169	8,995,153	7,864320	1,966,080	8,995,153	9,768,960	61,440	9,948,961
L3	6,635,520	1,143,081	983,040	245,760	1,143,081	1,213,440	15,360	1,258,641
L4	829,440	147,637	122,880	30,720	147,637	149,760	3,840	161,161

Five grid families have been generated to compute reference solutions for the hemisphere-cylinder benchmark configuration. Each family has at least four grid levels: L1 is the finest grid level, L2 is the second finest grid level, L3 is the next coarser level, etc. Statistics of grids from families 2 (tet), 4 (prism/hex), and 5 (struct) are shown in Table 3. Theses grids are used to compute the reference solutions in the accompanying paper [15].

Various metrics are used to characterize grid qualities. While there is no rigorously established relations between grid qualities and accuracy of the computed solutions (i.e., discretization errors), more regular grids (commonly considered as "higher-quality" grids) may reduce truncation errors and improve accuracy of some derived quantities, e.g., gradients. Regular grids are also preferable for some discretization methods (e.g., finite differences) and generally simplify numerical analysis. In many cases, mathematically rigorous statements about accuracy and convergence of discrete solutions are proved under assumption of regular grids.

In this section, we provide some quality characterization of grids generated for the special session at SciTech 2018. The following grid qualities are considered: (1) minimum and maximum cell volumes, (2) minimum planarity of quadrilateral faces, (3) minimum face skewness, and (4) the maximum cell closure defect.

To compute planarity of a quadrilateral face separating cells A and B, the face is divided by a diagonal cut into two triangular semi-faces. For each triangular semi-face, a directed area vector is computed, which is normal to the semi-face, points from cell A to cell B, and has the magnitude that is equal to the semi-face area. Planarity of a quadrilateral face is evaluated as cosine of the angle between the directed area vectors corresponding to these triangular semi-faces. Note that planarity is invariant with respect to the choice of the diagonal used to cut a quadrilateral face. The directed area of a quadrilateral face is the sum of the directed areas of the corresponding triangular semi-faces. A face with all its nodes placed in the same plane has the planarity value of positive one; any triangular face has planarity of one. The planarity value of zero corresponds to a quadrilateral face with orthogonal triangular semi-faces. The planarity value of negative one corresponds to a completely folded quadrilateral face.

Skewness of a face is evaluated as cosine of the angle between the face directed area and the edge vector. For the face directed area pointed from cell A to cell B, the edge vector points from the center of cell A to the center of cell B. The skewness value of positive one represents the edge vector perfectly orthogonal to the face; the skewness value of zero indicates that the edge is colinear with the face; and a negative skewness indicates that the edge vector and the face directed area point to opposite directions.

The cell closure defect is a sum of the face directed areas for a cell. Ideally, the sum of directed areas is expected to be zero, indicating a perfectly closed cell. Because of a round-off error, the computed defect is a non-zero value. Practical grids for aerospace applications often cover large computational domains. Cells near far extents of the domain can be quite large and produce significant defects. A large computational defect has many adverse implications. In particular, it strongly affects the "machine-zero" error level of a solution on a given grid.

Table 4: Hemisphere-cylinder grid characteristics.

Grid	Family 2 (tet)			Family 5 (struct)		
	Maximum Volume	Minimum Volume	Minimum Skewness	Maximum Volume	Minimum Volume	Minimum Skewness
L1	0.101	5.22×10^{-10}	7.87×10^{-4}	0.604	3.84×10^{-11}	0.511
L2	0.808	4.19×10^{-9}	7.94×10^{-4}	4.78	6.17×10^{-10}	0.503
L3	6.41	3.37×10^{-8}	8.06×10^{-4}	37.4	9.92×10^{-9}	0.492
L4	50.4	2.73×10^{-7}	8.29×10^{-4}	286	1.60×10^{-7}	0.488

Table 4 shows variation of cell volumes and skewness for hemisphere-cylinder grids of types 2 (tet) and 5 (struct). The minimum volume cells of family 5 (struct) grids are observed near the polar singularity; the minimum volume is reduced in grid refinement approximately by factor 16. The maximum volume of family 5 grids and the minimum and maximum volumes of unstructured grids of family 2 (tet) are reduced approximately by factor 8. A good minimum skewness of approximately 0.5 is observed for family 5 (struct) grids. As expected, tetrahedral grids of family 2 (tet) have much smaller minimum skewness of approximately 0.0008. Note that the minimum skewness is almost constant between grids in a family, indicating an excellent parametric similarity. Although not shown, the maximum cell defect observed on all grids used in the study does not exceed 10^{-13}. The defect decreases in grid refinement; the maximum defect on the finest family grids does not exceed 10^{-15}. An almost perfect face planarity is observed for all hemisphere-cylinder grids. The minimum face planarity among all grids of all families is observed for the structured L1 grid and is 0.9999999045.

V–B. OM6 Wing Grids Characteristics

Table 5: Statistics of grids for OM6 wing grid families.

Grid	Family 1 (prism)		Family 2 (tet)		Family 4 (prism/hex)		Family 5 (struct)	
	Cells	Nodes	Cells	Nodes	Cells	Nodes	Cells	Nodes
L1	121,110,528	60,777,345	363,331,584	60,777,345	69,206,016	60,777,345	69,206,016	69,305,025
L2	15,138,816	7,625,153	45,416,448	7,625,153	8,650,752	7,625,153	8,650,752	8,675,681
L3	1,892,352	960,225	5,677,056	960,225	1,081,344	960,225	1,081,344	1,087,665
L4	236,544	121,841	709,632	121,841	135,168	121,841	135,168	136,793

Similar to the hemisphere-cylinder configuration, five grid families have been generated for the OM6 wing configuration, including prismatic, tetrahedral, mixed-element, and structured grids. Statistics of grids from families 1 (prism), 2 (tet), 4 (prism/hex) and 5 (struct) are shown in Table 5. The largest grid of family 2 (tet) has more than 360 million cells. Table 6 presents some quality characteristics for grids of families 2 (tet) and 4 (prism/hex). In both families, the maximum cell volume decreases in grid refinement by almost exactly factor of 8. The minimum cell volume on grids of family 2 (tet) decreases regularly by a factor close to 32. The minimum cell volume reduction on grids of family 4(prism/hex) is not regular, the volume reduction factor varies from about 40 to less than 20. The minimum skewness and face planarity are small positive values on all grids. The minimum skewness measure is much smaller on tetrahedral grids than on mixed element grids composed from prismatic and hexahedral elements. For grids generated around non-trivial configurations, uniformly positive skewness and face planarity are considered as a sign of high quality. Negative skewness and face planarity are often observed on practical grids that diminish grid quality. Small variation of skewness and face planarity in grid refinement indicates close parametric similarity of grids in the same family.

Table 6: OM6 wing grid family characteristics.

Grid	Family 2 (tet)			Family 4 (prism/hex)			
	Maximum Volume	Minimum Volume	Minimum Skewness	Maximum Volume	Minimum Volume	Minimum Skewness	Minimum Face Planarity
L1	0.3	2.57×10^{-19}	1.15×10^{-5}	1.77	3.78×10^{-18}	0.0426	0.0664
L2	2.38	8.35×10^{-18}	1.31×10^{-5}	13.8	7.35×10^{-17}	0.0434	0.0555
L3	18.7	1.92×10^{-16}	1.22×10^{-5}	105	2.20×10^{-15}	0.0456	0.0492
L4	144	6.33×10^{-15}	1.36×10^{-5}	754	8.76×10^{-14}	0.0513	0.0432

VI. Input files and Grid Examples

The input file name is input.nml for the grid generation codes, and input_coarsen.nml for the grid coarsening code. The following input file for the hemisphere-cylinder grid generator generates a half-geometry grid of family 2 (tet) with 63,813 nodes and 354,816 tetrahedral elements shown in Figure 7:

```
&input_parameters
  target_reynolds_number  =  350000
           target_y_plus  =  1.0
              igrid_type  =  2
                 R_outer  =  100
  nnodes_cylinder_input   =  24
                   nr_gs  =  8
                     nre  =  88
              domain_cut  =  2
     generate_ugrid_file  =  T
  ugrid_file_unformatted  =  T
         generate_k_file  =  T
      k_file_unformatted  =  F
   generate_line_file_nc  =  T
   generate_line_file_cc  =  T
     generate_tec_file_b  =  T
     generate_tec_file_v  =  F
/
```

Subsequently, the following input file for the grid coarsening code generates three coarse grids shown in Figures 8-10:

```
&input_parameters
                   project  =  hc_tetra
                 tet2prism  =  F
    ugrid_file_unformatted  =  T
        k_file_unformatted  =  F
      rmp_file_unformatted  =  T
  prolong_file_unformatted  =  T
       generate_tec_file_b  =  T
       generate_tec_file_v  =  F
/
```

For the OM6 wing, the following example input file is provided:

```
&input_parameters
          target_reynolds_number  =  14.6e6
                   target_y_plus  =  1.0
                      igrid_type  =  2
                    airfoil_type  =  0
               airfoil_data_file  =  "om6_wing_section_sharp.dat"
                              tr  =  0.5625159852668158
                            beta  =  29.9990
                         R_outer  =  100
          nnodes_cylinder_input   =  32
                               b  =  1.476017976219800
                           nr_gs  =  8
                             nre  =  64
                       wing_side  =  2
                       root_le_x  =  0.0
    stretching_tanh_towards_lete  =  4.5
             generate_ugrid_file  =  T
          ugrid_file_unformatted  =  T
                 generate_k_file  =  T
              k_file_unformatted  =  F
           generate_line_file_nc  =  T
           generate_line_file_cc  =  T
             generate_tec_file_b  =  T
             generate_tec_file_v  =  F
/
```

This input file generates a type 2 (tet) grid with 113,945 nodes and 663,552 tetrahedral elements shown in Figure 11. Subsequently, the following input file for the grid coarsening code generates three coarse grids shown in Figures 12-14:

```
&input_parameters
                    project  =  wing_tetra
                  tet2prism  =  F
    ugrid_file_unformatted  =  T
        k_file_unformatted  =  F
      rmp_file_unformatted  =  T
   prolong_file_unformatted  =  T
        generate_tec_file_b  =  T
      generate_tec_file_vol  =  F
/
```

VII. Concluding Remarks

Customized grid generation and coarsening codes have been developed for benchmark 3D configurations used for a special session at AIAA SciTech 2018 conference. The configurations are a hemisphere cylinder and the ONERA M6 wing. The codes allow generation of high-quality consistent nested grid families. The grids can include various elements such as tetrahedra, prisms, and hexahedra. The finest grids in the families have up to 400 million elements. Five families of grids have been generated, four unstructured-grid families and one structured-grid family. Many of these grids are used for studying grid convergence of reference solutions for complex turbulent flows around the benchmark configurations. Consistency of grid refinement and close parametric similarity of grids of the same family are critical factors enabling accurate grid-converged solutions that can be used for verification of advanced CFD solvers for turbulent flows.

Acknowledgments

The NASA Aeronautics Research Mission Directorate Transformative Aeronautics Concepts Program Transformative Tools and Technologies project supported the work reported herein. The first author acknowledges support by NASA under Contract No. NNL09AA00A.

References

[1] AIAA, "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," AIAA G-077-1998, 1998.

[2] Oberkampf, W. L. and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, 2010.

[3] "Drag Prediction Workshop," `aiaa-dpw.larc.nasa.gov`, visited on 12/05/2017.

[4] "High Lift Prediction Workshop," `hiliftpw.larc.nasa.gov`, visited on 12/05/2017.

[5] "GridEX," `https://geolab.larc.nasa.gov/GridEx`, visited on 12/05/2017.

[6] Parikh, P., Pirzadeh, S., and Löhner, R., "A Package for 3-D Unstructured Grid Generation Finite-Element Flow Solution and Flow Field Visualization," NASA CR–182090, 1990.

[7] Pirzadeh, S. Z., "Advanced Unstructured Grid Generation for Complex Aerodynamic Applications," *AIAA J.*, Vol. 48, No. 5, 2010, pp. 904–915.

[8] "Pointwise, Inc." `http://www.pointwise.com`, visited on 12/05/2017.

[9] "Turbulence Modeling Resource Website," `http://turbmodels.larc.nasa.gov`, visited on 12/05/2017.

[10] Rumsey, C. L., Smith, B. R., and Huang, G. P., "Description of a Website Resource for Turbulence Modeling Verification and Validation," AIAA Paper 2010-4742, 2010.

[11] Tsieh, T., "An Investigation of Separated Flow About a Hemisphere-Cylinder at 0- to 19-Deg Incidence in the Mach Number Range from 0.6 to 1.5," AEDC-TR-76-112, 1976.

[12] Gleize, V., Dumont, A., Mayeur, J., and Destarac, D., "RANS Simulations on TMR Test Cases and M6 Wing with the ONERA elsA Flow Solver," AIAA Paper 2015-1745, 2015.

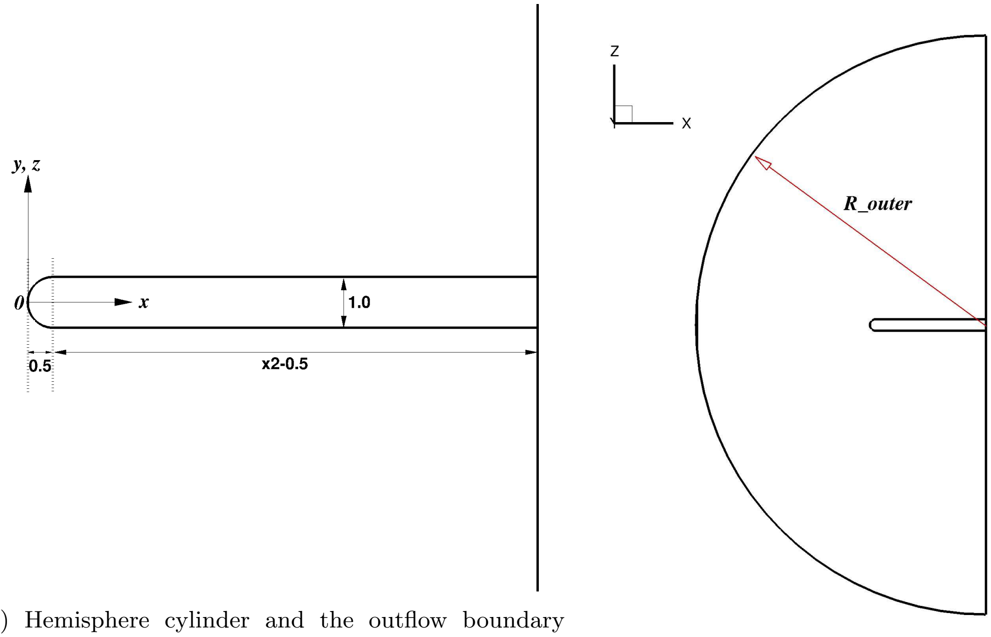
[13] Mayeur, J., Dumont, A., Gleize, V., and Destarac, D., "RANS Simulations on TMR 3D Test Cases with the ONERA elsA Flow Solver," AIAA Paper 2016-1357, 2016.

[14] Dumont, A., Mayeur, J., Dumont, A., Destarac, D., and Gleize, V., "Reynolds-Averaged Navier-Stokes Simulations on NACA0012 and ONERA-M6 Wing with the ONERA elsA Solver," *AIAA J.*, Vol. 54, No. 9, September 2016, pp. 2671–2687.

[15] Diskin, B., Anderson, W. K., Pandya, M. J., Rumsey, C. L., Thomas, J. L., Liu, Y., and Nishikawa, H., "Grid Convergence for Three Dimensional Benchmark Turbulent Flows," SciTech-2018, Kissimmee, FL, Jan., 2018, To be published as AIAA Paper.

[16] White, F. M., *Fluid Mechanics*, McGraw-Hill, seventh ed., 2011.

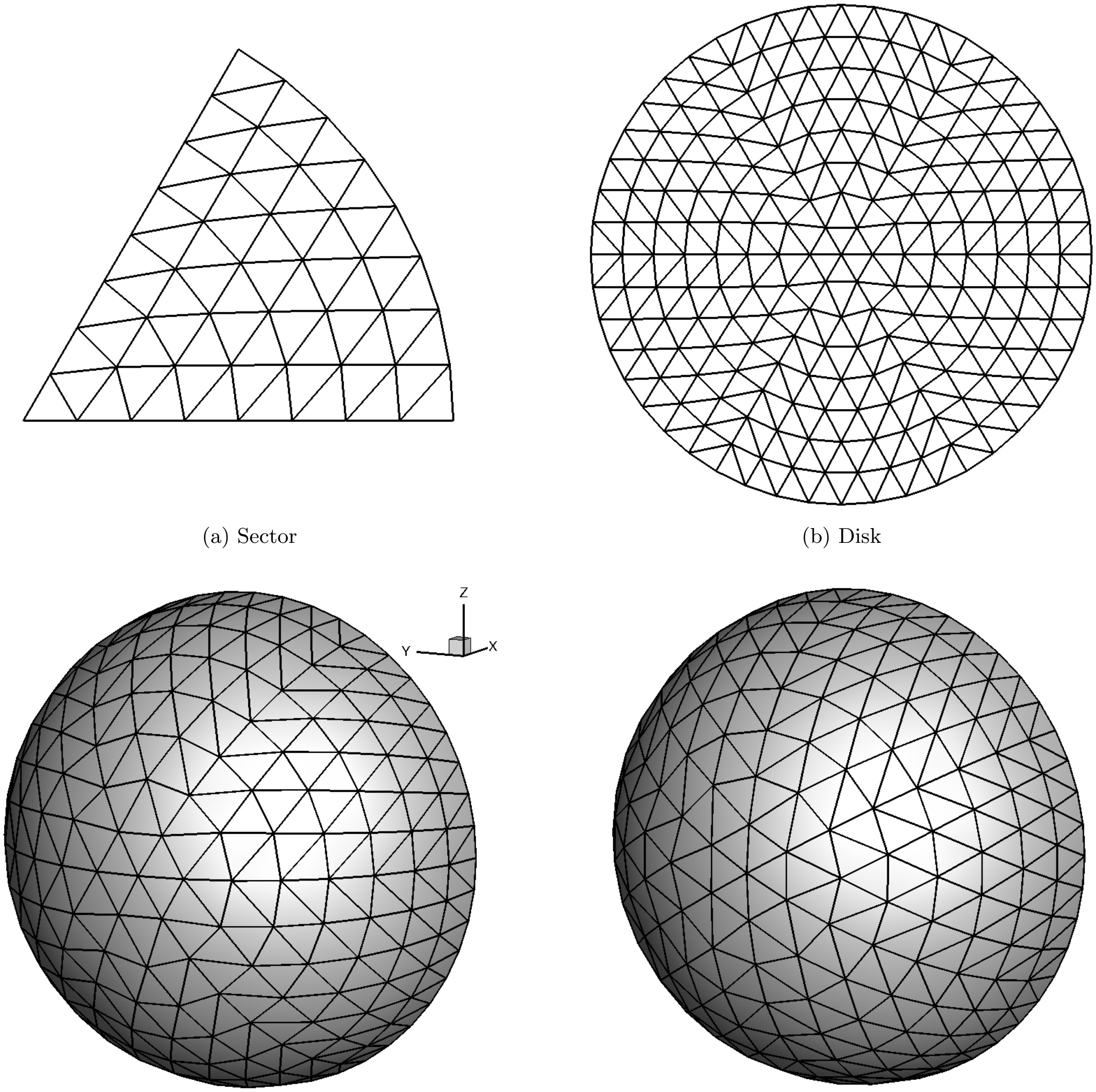
[17] White, F. M., *Viscous Fluid Flow*, McGraw-Hill, 2nd ed., 1991.



(a) Hemisphere cylinder and the outflow boundary plane.

(b) Computational domain.

Figure 1: Hemisphere cylinder geometry and computational domain. The hemisphere cylinder has the radius 0.5 and the length is defined by x2, where x2 is a user input parameter. The outer boundary is a hemisphere centered at $(x, y, z) = (\texttt{x2}, 0, 0)$ with the radius R_outer, where R_outer is another user input parameter.



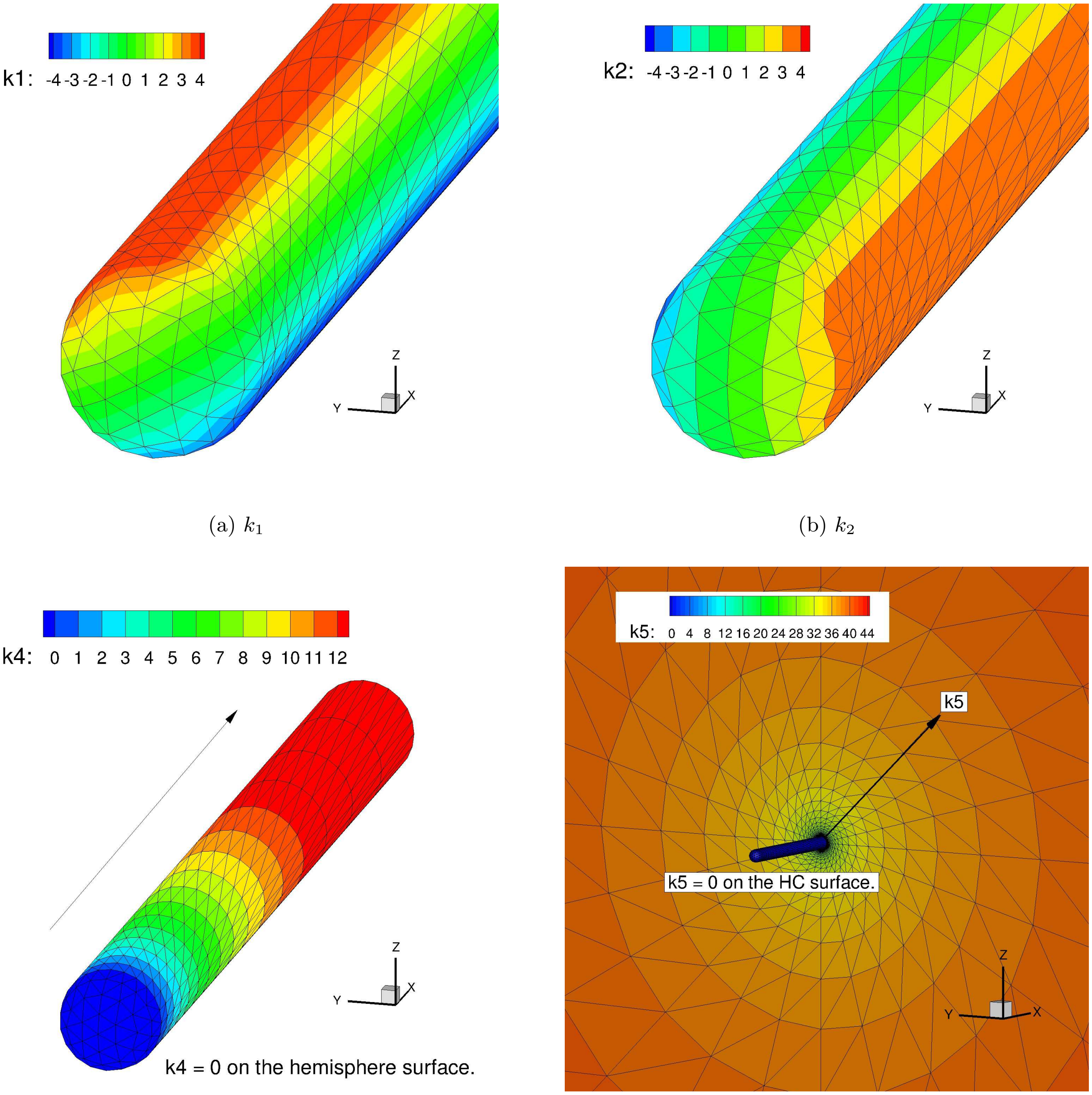
(a) Sector

(b) Disk

(c) Hemisphere

(d) Rotated hemisphere

Figure 2: Construction of the hemisphere surface mesh. Example is shown for `nr_gs` = 8.


k1: -4 -3 -2 -1 0 1 2 3 4
Z
Y
X
k2: -4 -3 -2 -1 0 1 2 3 4
Z
Y
X
k4: 0 1 2 3 4 5 6 7 8 9 10 11 12
k4 = 0 on the hemisphere surface.
Z
Y
X
k5: 0 4 8 12 16 20 24 28 32 36 40 44
k5
k5 = 0 on the HC surface.
Z
Y
X

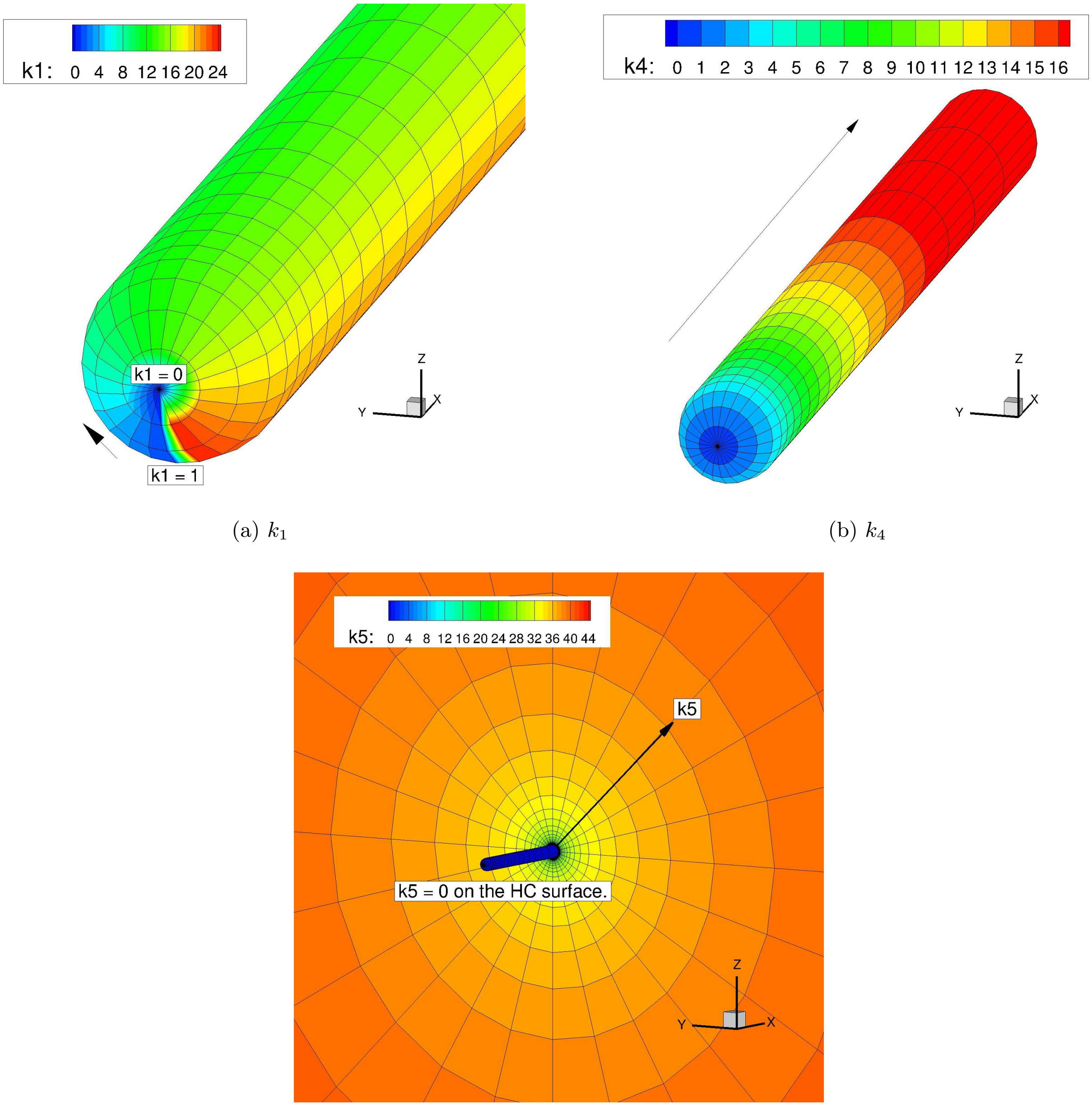
(a) k_1

(b) k_2

(c) k_4

(d) k_5

Figure 3: Indices k_1, k_4, and k_5 that represent unstructured grids. $k_1 \in [-\texttt{nr\_gs}, \texttt{nr\_gs}]$, $k_2 \in [-\texttt{nr\_gs}, \texttt{nr\_gs}]$, $k_4 \in [0, \dots, \texttt{nnodes\_cylinder\_input}]$, and $k_5 \in [0, \texttt{nre}]$. $k_3 = -(k_1 + k_2)$ is not shown. Examples are shown for $\texttt{nr\_gs} = 4$, $\texttt{nnodes\_cylinder\_input} = 12$, and $\texttt{nre} = 44$.


k1: 0 4 8 12 16 20 24
k1 = 0
k1 = 1
Z
Y
X
k4: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Z
Y
X
k5: 0 4 8 12 16 20 24 28 32 36 40 44
k5
k5 = 0 on the HC surface.
Z
Y
X

(a) k_1

(b) k_4

(c) k_5

Figure 4: Indices k_1, k_4, and k_5 that represent structured grids. $k_1 \in [1, 6 \times \texttt{nr\_gs}]$, $k_4 \in [0, \texttt{nr\_gs} + \texttt{nnodes\_cylinder\_input}]$, and $k_5 \in [0, \texttt{nre}]$. Examples are shown for $\texttt{nr\_gs} = 4$, $\texttt{nnodes\_cylinder\_input} = 12$, and $\texttt{nre} = 44$. k_2 and k_3 are not used in the structured grid.

Downloaded by Hiroaki Nishikawa on January 8, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-1101

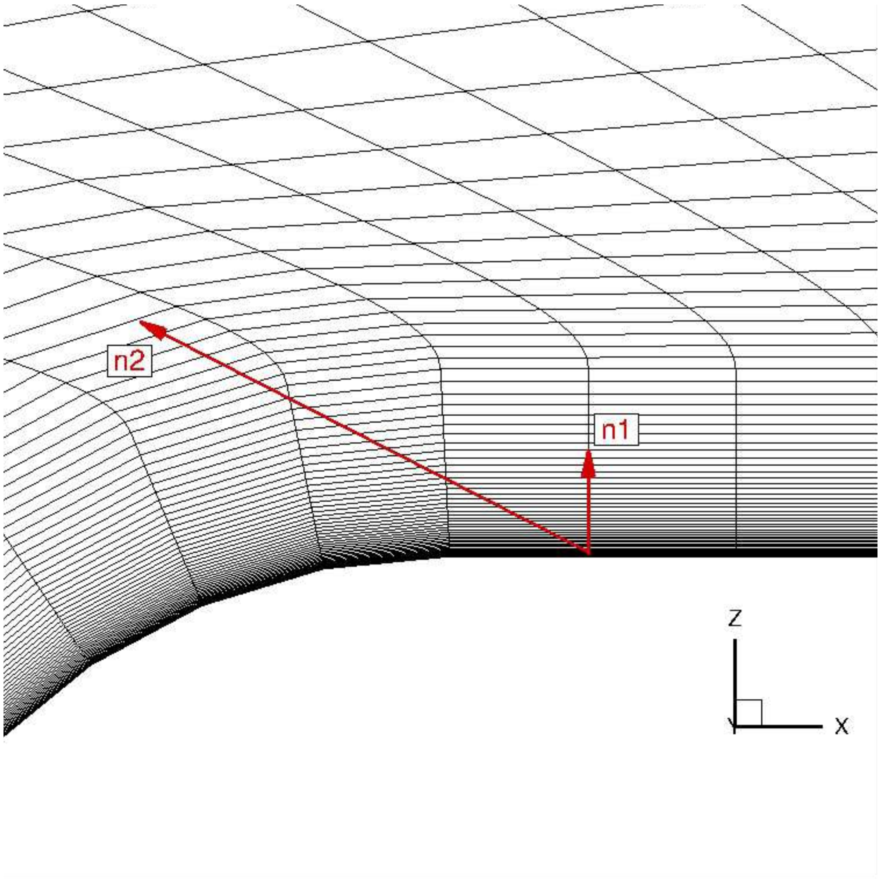
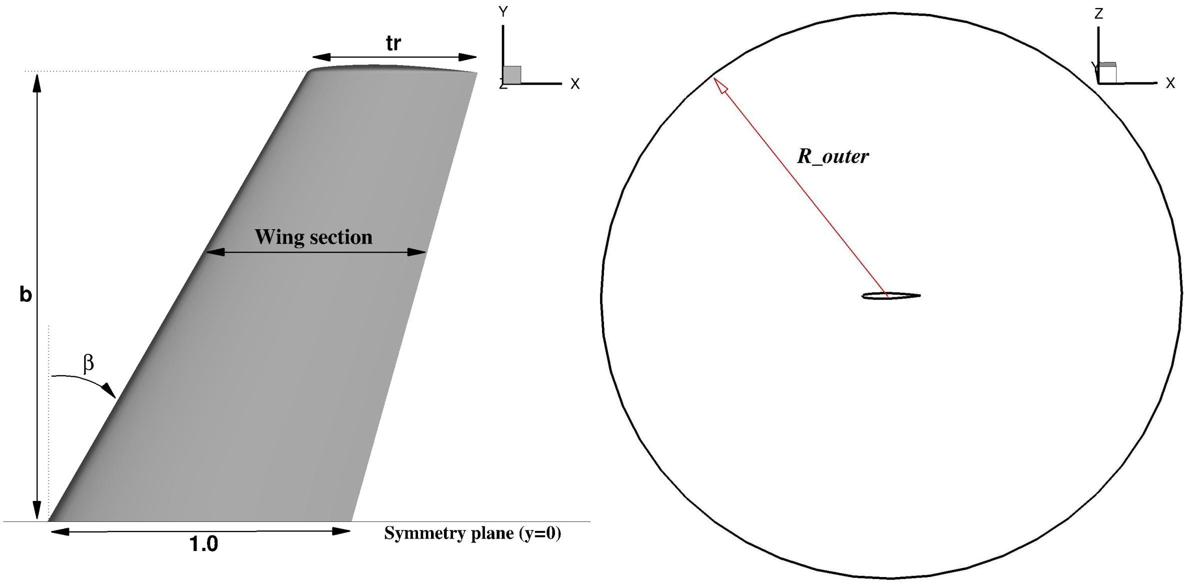


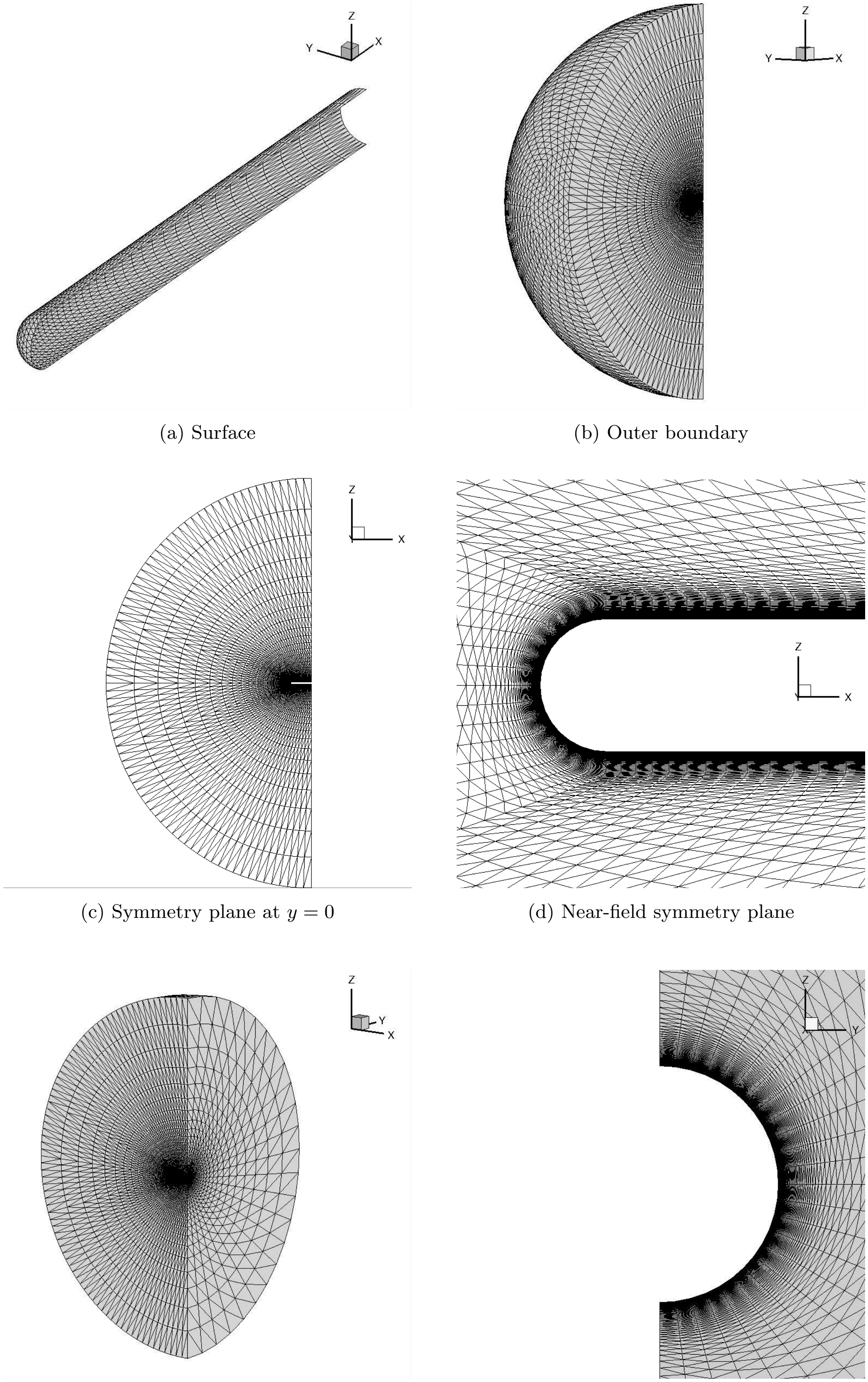
Figure 5: Two normals blended to give the direction for node generation from a surface node to the corresponding outer boundary node. $\mathbf{n}_1$ is the surface normal, and $\mathbf{n}_2$ is the direction straight from the surface node to the outer boundary node.



(a) OM6 wing and the symmetry plane.

(b) Computational domain.

Figure 6: OM6 wing geometry and computational domain. This is the right-wing configuration. The wing and the outer boundary (hemisphere) are located in the region $y \geq 0$ (the other side of the paper). The leading edge is at the origin.



(a) Surface

(b) Outer boundary

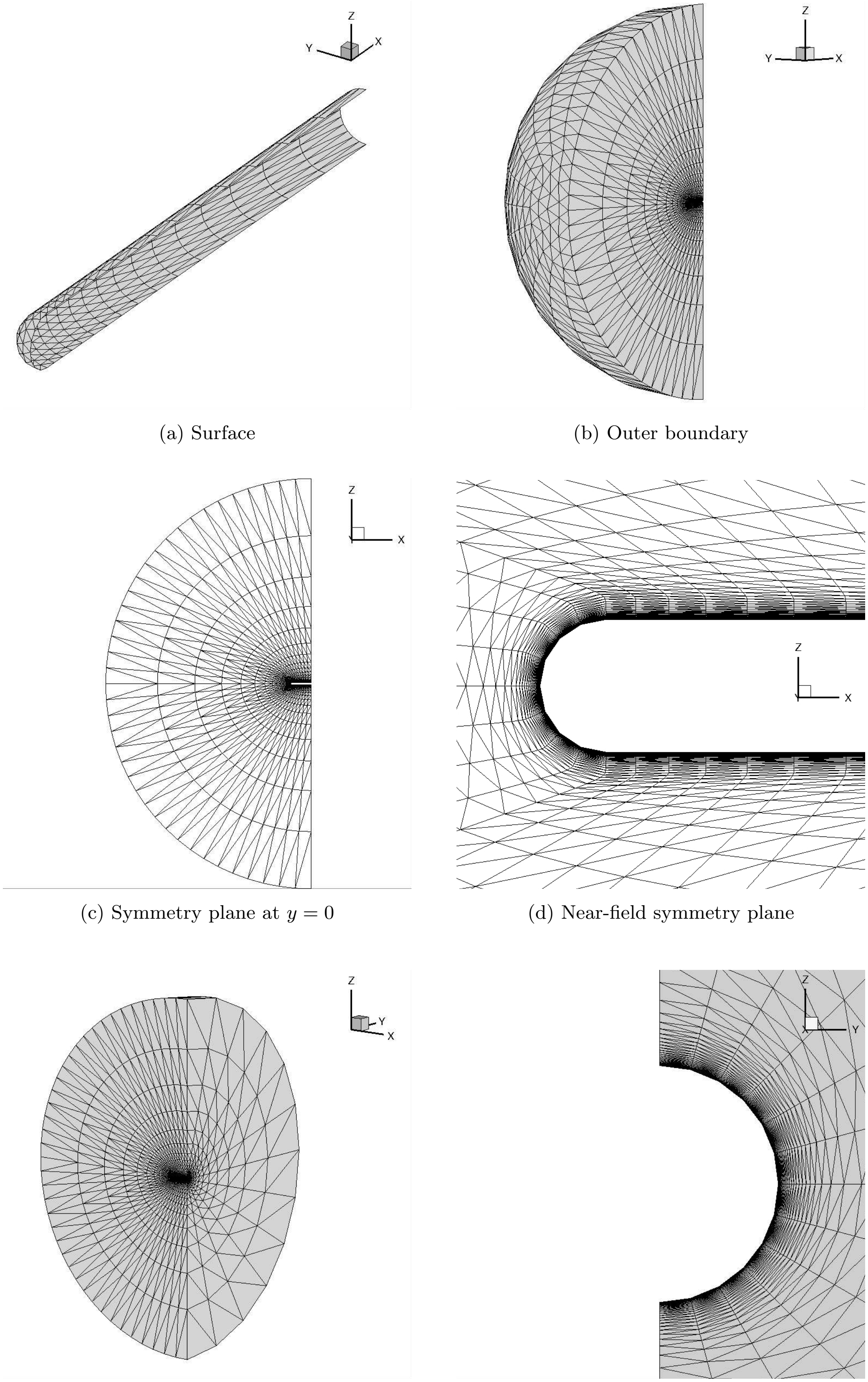
(c) Symmetry plane at $y = 0$

(d) Near-field symmetry plane

(e) Outflow boundary

(f) Near-field outflow

Figure 7: Level 1 tetrahedral grid for the hemisphere cylinder (half geometry).



(a) Surface

(b) Outer boundary

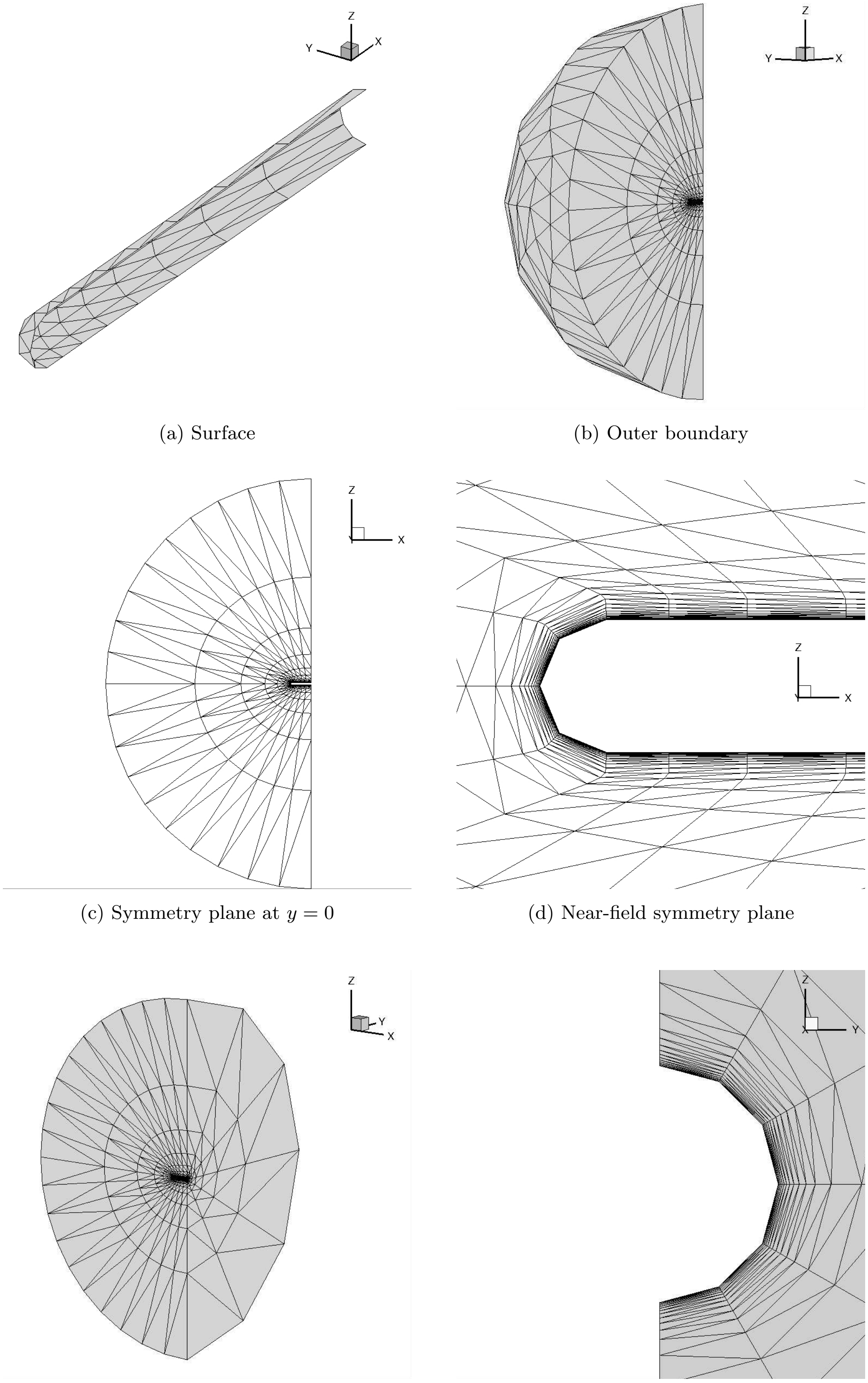
(c) Symmetry plane at $y = 0$

(d) Near-field symmetry plane

(e) Outflow boundary

(f) Near-field outflow

Figure 8: Level 2 tetrahedral grid for the hemisphere cylinder (half geometry).



(a) Surface

(b) Outer boundary

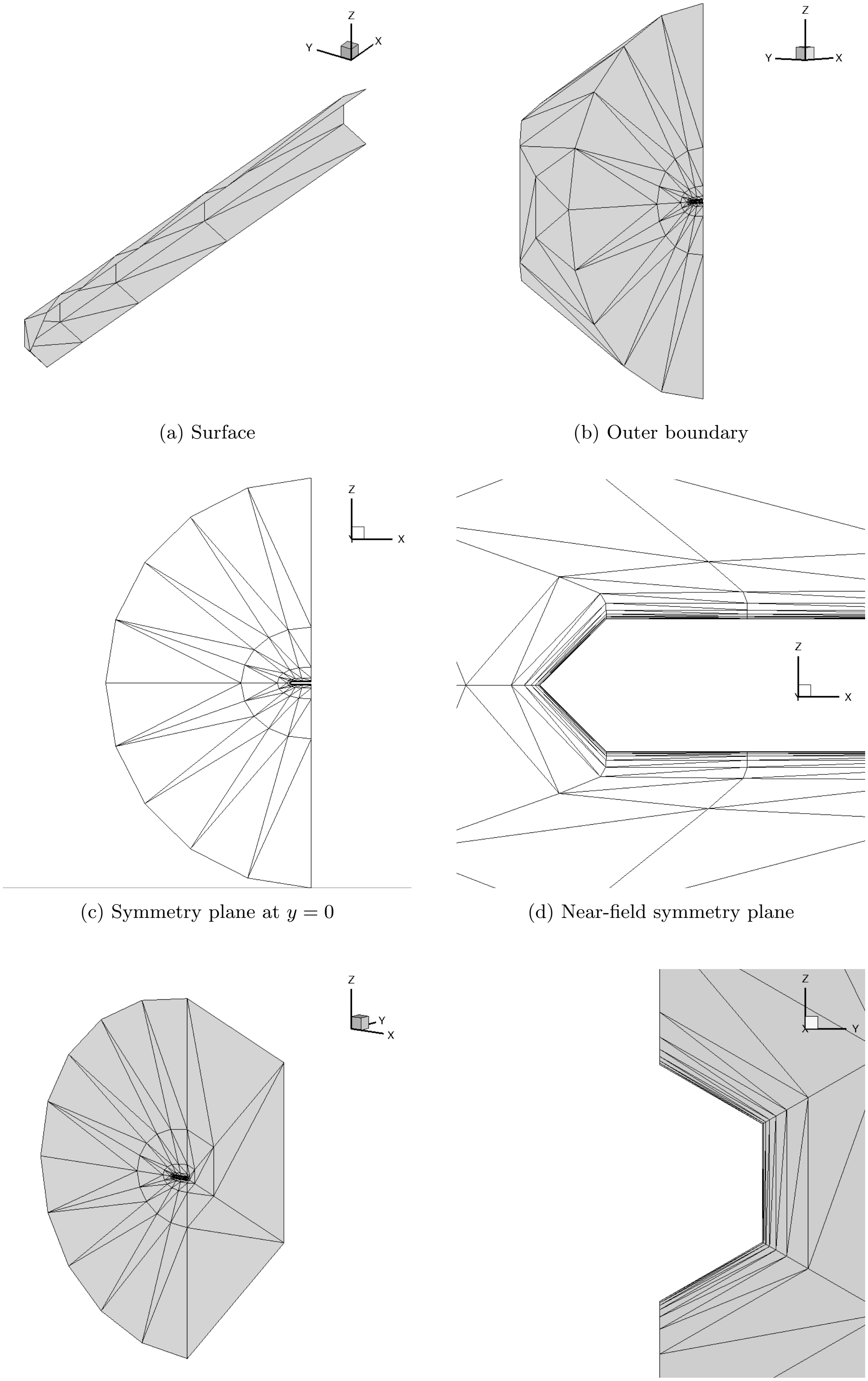
(c) Symmetry plane at $y = 0$

(d) Near-field symmetry plane

(e) Outflow boundary

(f) Near-field outflow

Figure 9: Level 3 tetrahedral grid for the hemisphere cylinder (half geometry).



(a) Surface

(b) Outer boundary

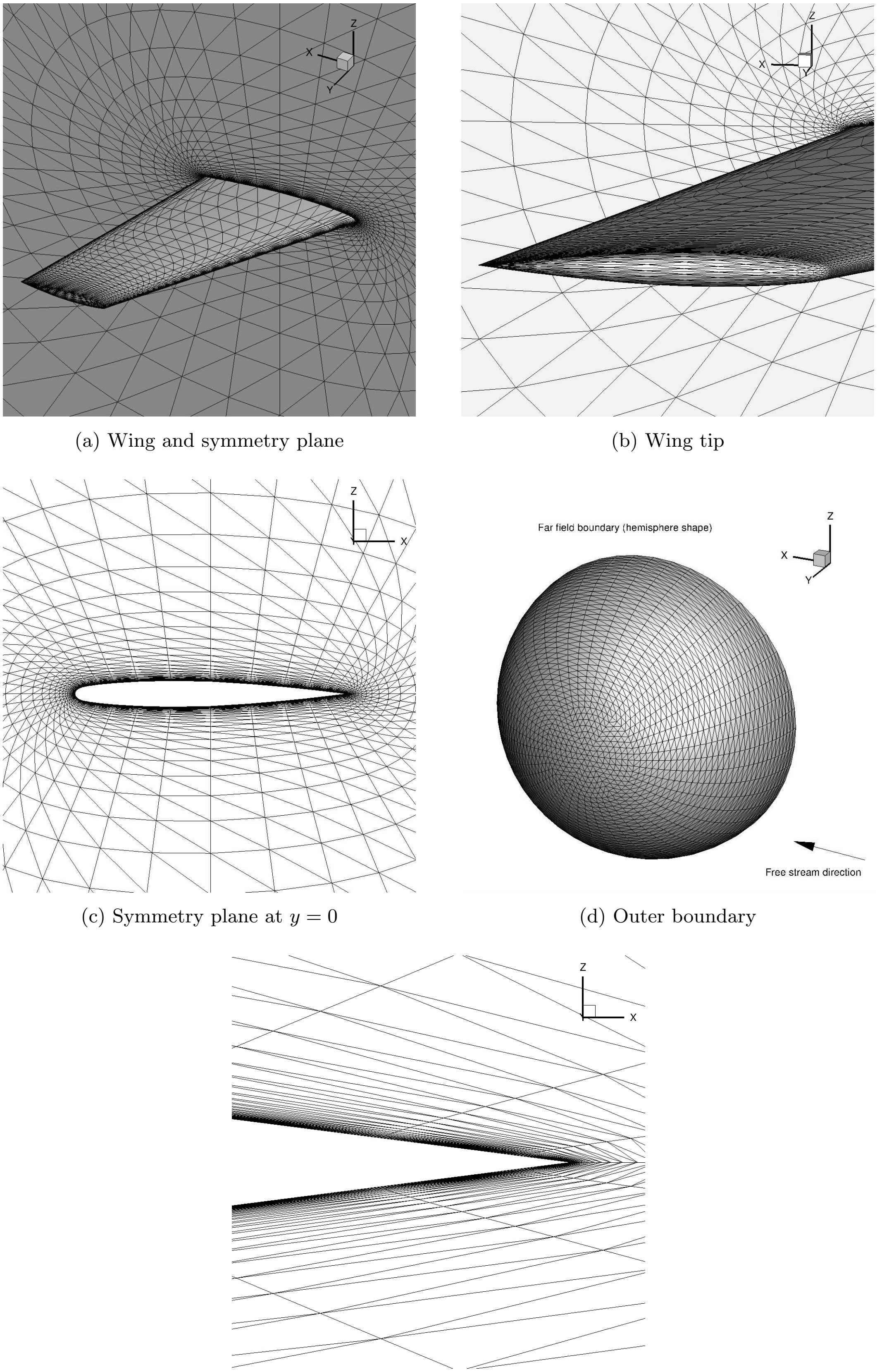
(c) Symmetry plane at $y = 0$

(d) Near-field symmetry plane

(e) Outflow boundary

(f) Near-field outflow

Figure 10: Level 4 tetrahedral grid for the hemisphere cylinder (half geometry).



(a) Wing and symmetry plane

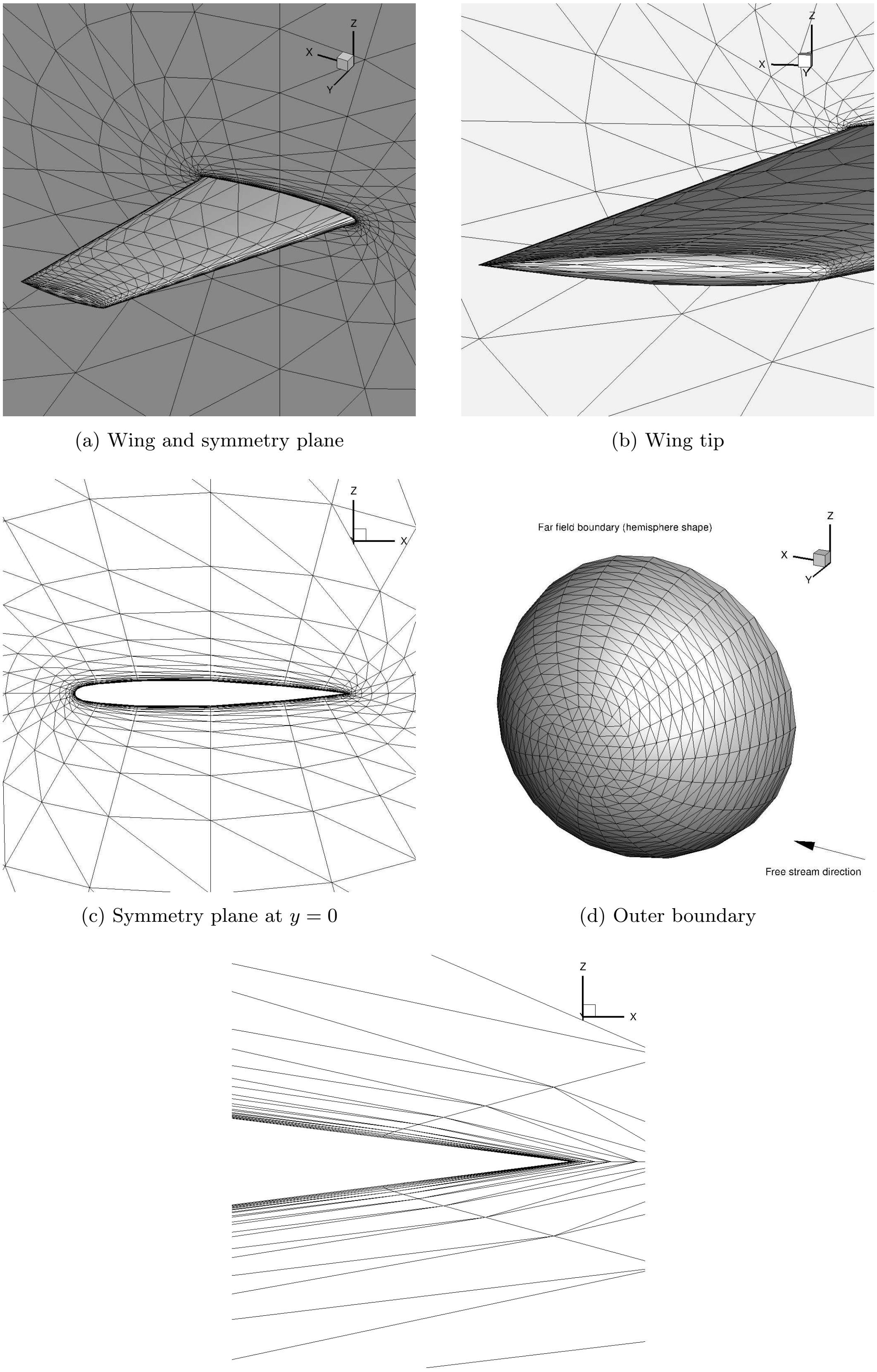
(b) Wing tip

(c) Symmetry plane at $y = 0$

(d) Outer boundary

(e) TE region

Figure 11: Level 1 tetrahedral grid for the OM6 wing.



(a) Wing and symmetry plane

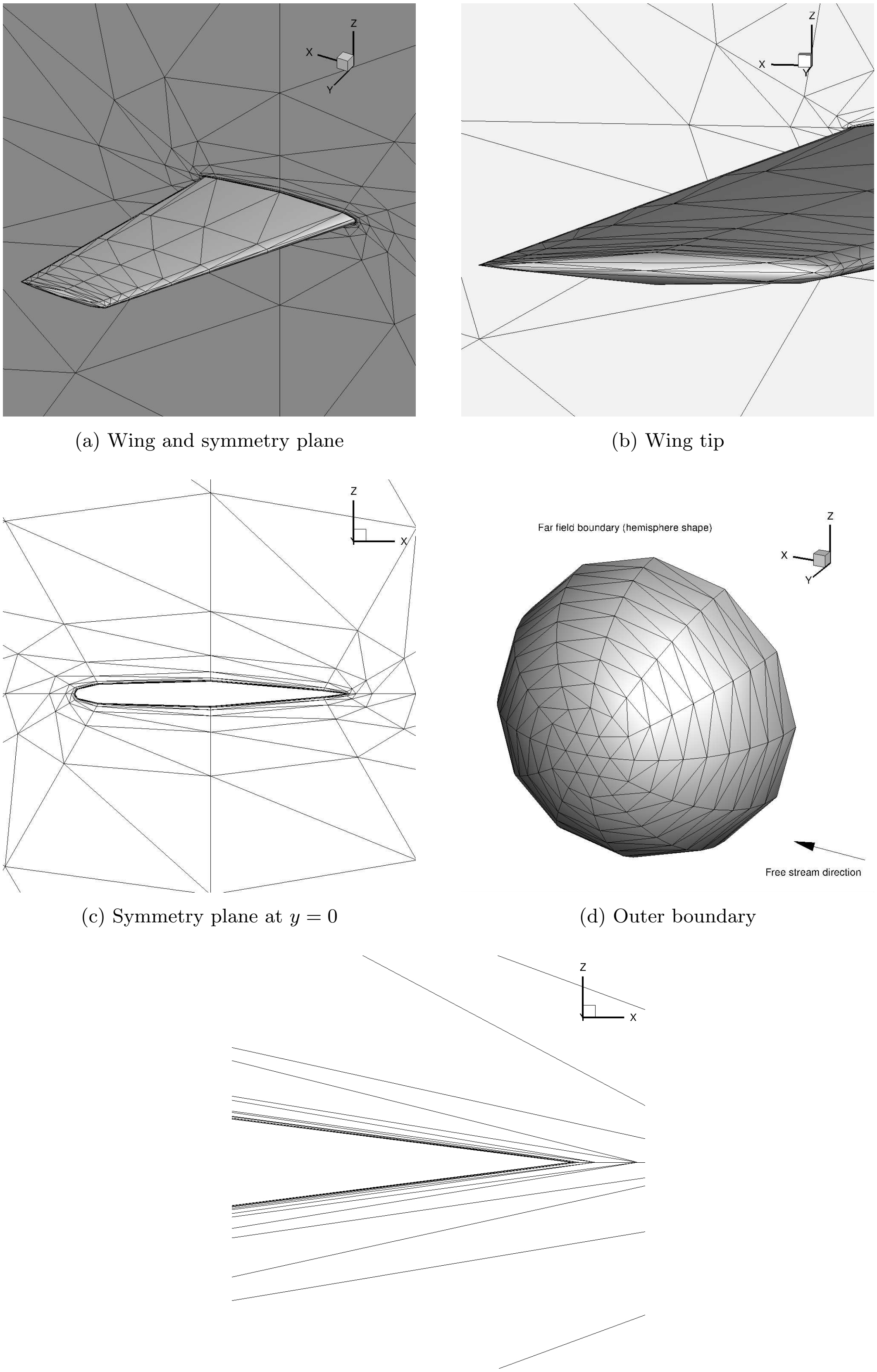
(b) Wing tip

(c) Symmetry plane at $y = 0$

(d) Outer boundary

(e) TE region

Figure 12: Level 2 tetrahedral grid for the OM6 wing.



(a) Wing and symmetry plane

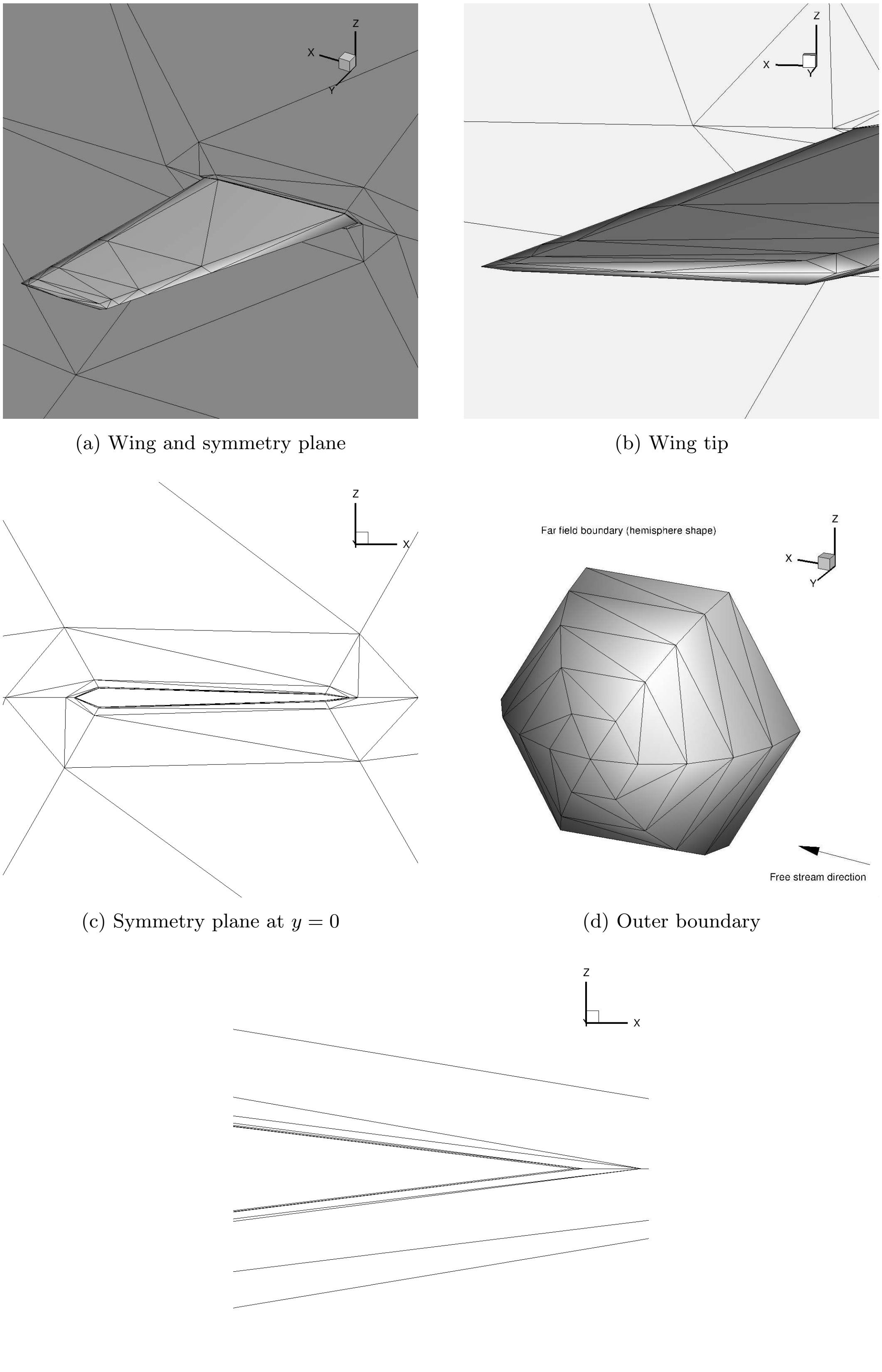
(b) Wing tip

(c) Symmetry plane at $y = 0$

(d) Outer boundary

(e) TE region

Figure 13: Level 3 tetrahedral grid for the OM6 wing.



(a) Wing and symmetry plane

(b) Wing tip

(c) Symmetry plane at $y = 0$

(d) Outer boundary

(e) TE region

Figure 14: Level 4 tetrahedral grid for the OM6 wing.